Introduction

Our project provides a setup for detecting sound remotely by reflecting a laser beam off a hard surface, usually a window. Any sound that is near a window causes the window to move, and this technology takes advantage of that. It allows sounds to be heard from very far away because the sound information travels using the light as a medium, instead of the pressure waves of sound, it attenuates much less quickly. Another interesting thing to note is that the sound information is traveling at the speed of light instead of the speed of sound, so the information arrives more quickly than it would in a normal situation.

As one might imagine, this has interesting surveillance applications. This technology is currently being used by the CIA and many other surveillance-related organizations to eavesdrop. However, the main difference is that they use phase detection and infrared lasers while we use amplitude detection and a ruby laser for cost purposes. The phase modulation is much more accurate and much less noise prone, but requires a more complicated setup and was also not the goal of our project. The infrared laser is also useful in real-world scenarios because of the fact that it is invisible; the ruby laser might cause the surveillance subject to realize that they are being watched.

Setup

The setup itself is rather simple. It just consists of a laser pointer that is pointed at a window or any reflective hard surface. The sound vibrations cause the hard surface to act as a diaphragm and vibrate along with the sound. There is also a photodetector that picks up the light and measures the intensity, which is sent to a computer. We used the audio-in line on a laptop to input the changes in voltage measured by the photodetector to the computer. Then, on the computer we performed all our signal processing through Matlab 7.0 and Labview.

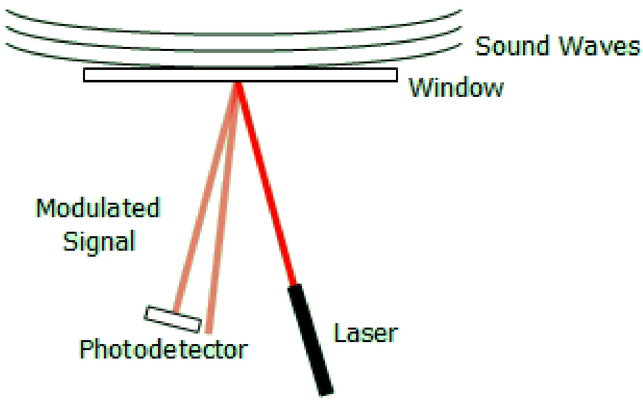
This vibration in the reflective diaphragm causes the laser beam to change direction slightly, which causes the intensity that is perceived by the photodetector to change. Our first laser pointer was more focused and would cause our photodetector to maximize its output (causing railing or clipping) which would make changes undetectable. To rectify this situation we moved the laser beam slightly off the photodetector so that it was only partially hitting. Causing it to rail then moving it slightly off the photodetector resulted in the best sounding signal. The resulting changes in intensity are then sent through the audio line.

There are several problems that must be dealt with in the implementation of this laser microphone that are listed as follows:

1. In addition to the laser light, ambient light is picked up by the photodetector. This ambient light may change and vary randomly, or may be synched with the 60Hz frequency of the electrical grid if the lights are florescent, which most of them are. Most of the ambient light was removed by simply adding a dark long tube for the laser to pass through before it reached the photodiode at the end. Implementing this blocks out a large percentage of the light, as it is not aligned directly with the tube and therefore cannot reach the photodiode. The back side of the tube must also be protected from light, so an opaque cloth covering was used to allow the wires attached to the photodiode to have freedom of movement.
2. There exists basic electrical noise on the circuit. This noise comes from both fluorescent lights, and from EMF noise produced by the

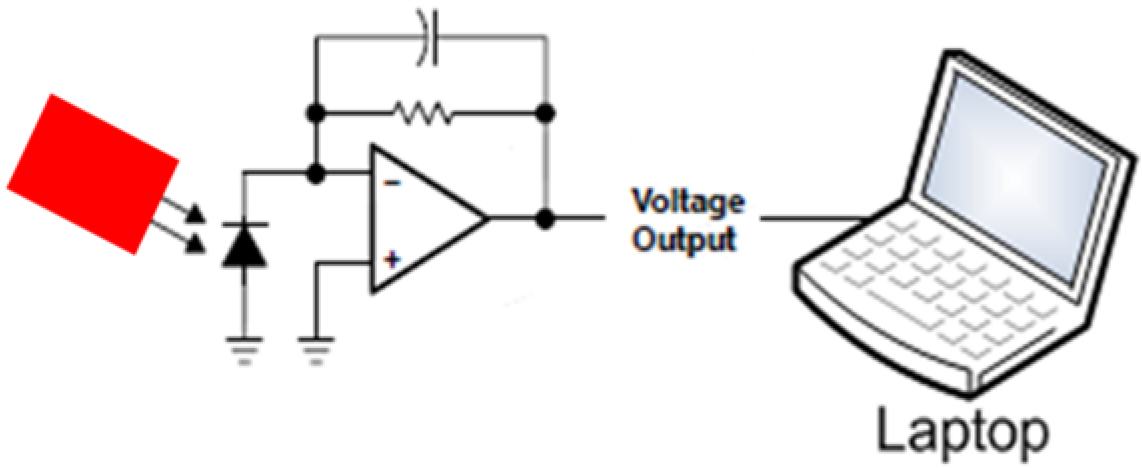
power grid being picked up on the wires. The noise is at 60Hz, 120Hz, and other harmonics of 60Hz.

3. most importantly, there are significant changes in the sound signal due to the properties of the window. The window has properties such as the size, thickness, and the choice of material. These properties alter how the window vibrates when it receives the sound signal from the air. The window can be treated as a filter to the sound, as it resonates with certain frequencies and dampens others. We solved this problem with a complex set of inverse filters that will be explained in detail later in this document.



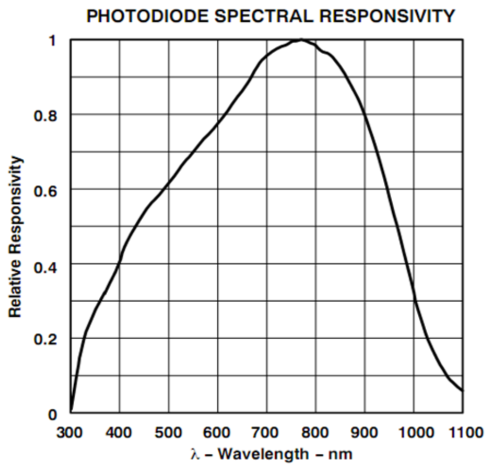
Implementation

The hardware implementation of the Laser Microphone is relatively simple and can be done at a minimal cost.



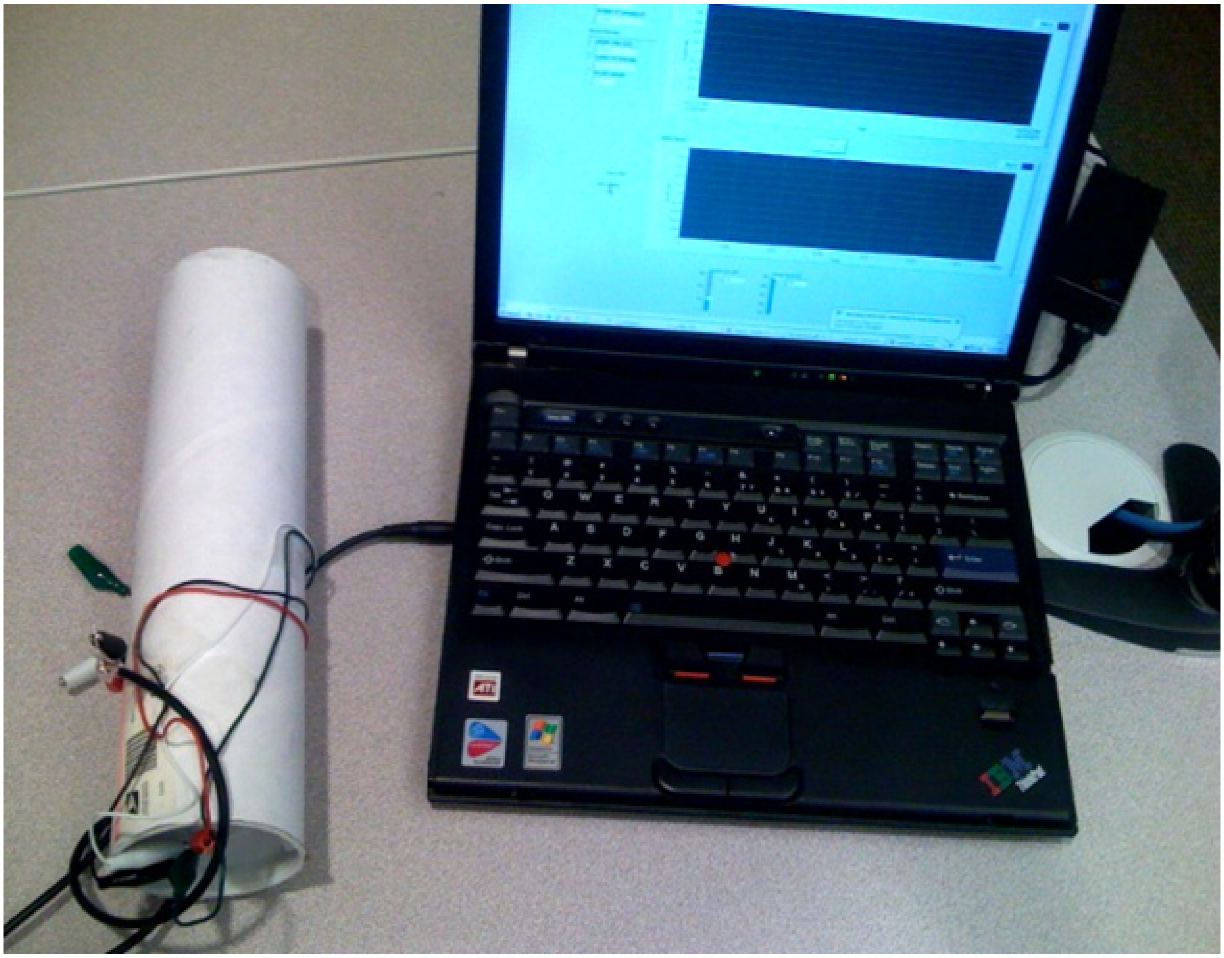
Laser/Photodetector:

The laser we used was a simple presentation laser pointer that outputted a red beam at approximately a 650nm wavelength. To receive the signal, we used a low cost photodetector (TSL12s), which is simply a photodiode and a trans-impedance amplifier combined together in a single package. The peak of the photodetector's spectral response characteristics coincide with the output wavelength of the laser pointer.



Detection Unit:

The laser capture setup was a cardboard tube with a small hole in one end for a photodetector in order to obstruct as much ambient light as possible. A power supply was used to create the 5 volt supply voltage for the photodetector and a spliced 1/8” phono jack connecter was connected to the outputted signal.



DAC:

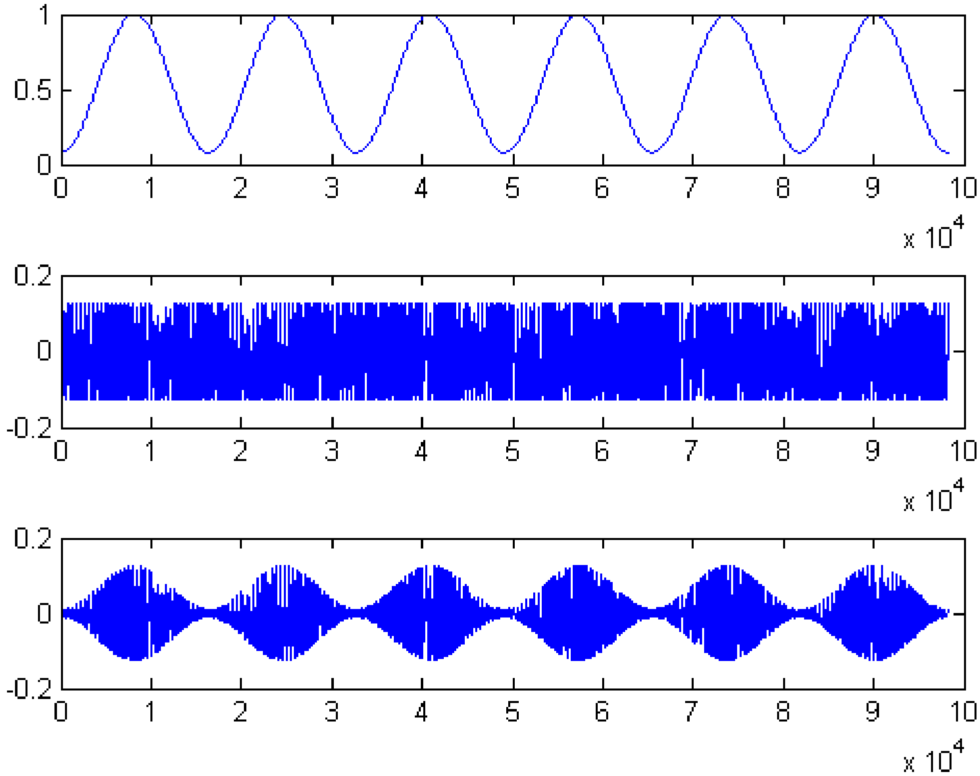
The Digital to Analog converter that we use to digitize the signal for further software processing is the mic-in jack on a laptop. Using this 22.05 kHz DAC, we are able to cheaply and properly sample the 3.6 kHz speech signal while following the Nyquist criterion and thus avoid any aliasing effects.

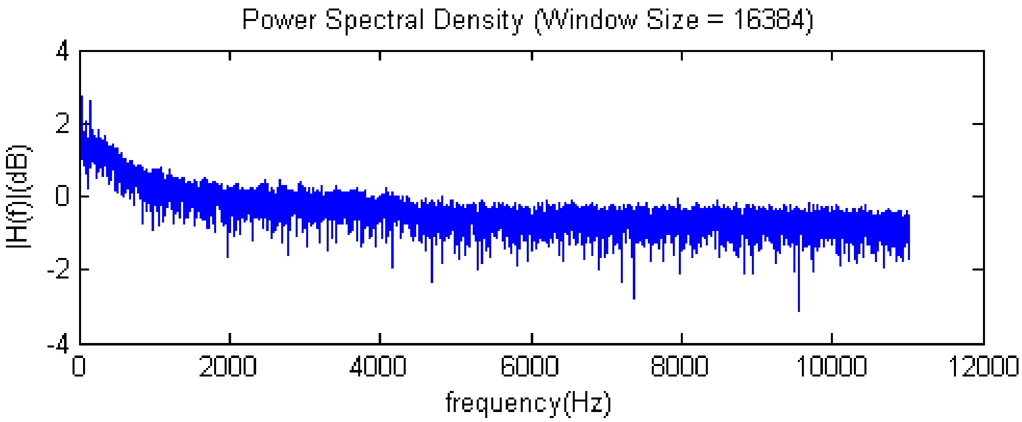
Inverse Filter

We observed that the system did not transmit sound information perfectly, and transmitted speech signals suffered some distortion. This distortion happens for two reasons: (1) the physical properties of the glass cause it to respond differently to different frequencies, and (2) low-frequency vibrations caused by air-conditioning systems and other building vibrations are constantly present in the window. We attempted to compensate for this observed distortion by building an inverse filter. We accomplished this in three steps:

Step 1: Measure the Frequency Response

In order to accurately model the system, we needed to measure its frequency response. We blasted a 30-second sound clip of pure white noise at the window and recorded the signal measured by the detection unit. Since we knew the input of the system (the white noise) had a completely flat spectrum, the output's spectrum should represent the frequency response. To compute the spectrum of the output (the recorded signal), we windowed portions of the signal using a Hamming window, computed the FFT's of each windowed portion, and then averaged the FFT's. This average FFT represents the frequency response of our system.





The plot shows some strong low-frequency vibrations in the window. We attributed these to the air-conditioning unit in the building and to other random vibrations in the environment. We also noticed that the window responded better to low frequencies than to high frequencies. This could be

a result of the physical properties of the glass as well as the physical dimensions of the window.

Step 2: Model the System

Once we had a good idea of the system's frequency response, we attempted to model the system using a linear prediction filter. We used a linear prediction filter because it made the inverse filter simple to implement, and it guaranteed that the inverse filter would be inherently stable and have a linear phase response. A linear prediction filter estimates its next output by the current input and a linear combination of n previous outputs:

$$y[n] = \; g \cdot x[n] + \sum_{k=1}^{N} a_k \cdot y[n-1]$$

The first step to building this filter is to compute the autocorrelation coefficients of the recorded signal. The autocorrelation coefficients are a measure of the correlation between samples of the signal. Since the filter must accurately estimate the output based on previous outputs, it must preserve the correlation between samples. One autocorrelation coefficient r[i] can be expressed as:

$$r_i = \; \sum_{n=0}^{N-i+1} s[n]s[n+i]$$

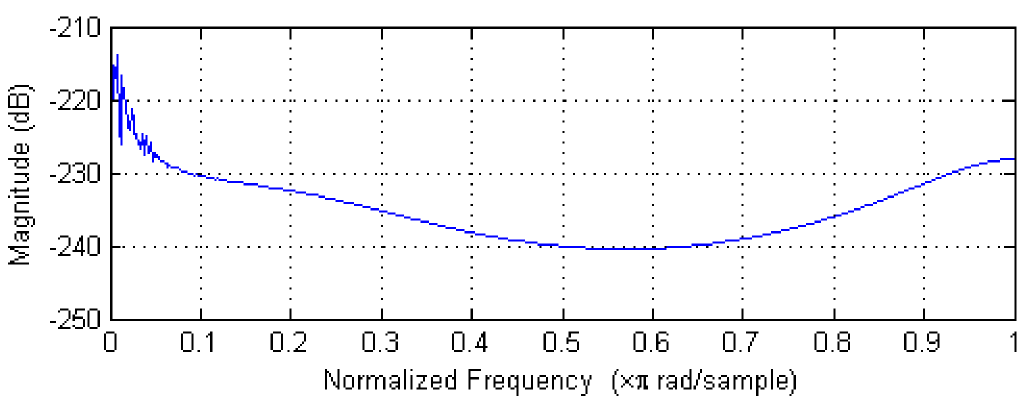
The next step to building the filter was to compute the filter coefficients. We used a recursive algorithm called Burden's Algorithm to do this. We set the first coefficient a[0] = 1 and then compute the other coefficients recursively:

$$K_i = -\frac{r_i + a_1 \cdot r_{i-1} + \cdots + a_{i-1} \cdot r_i}{E_{i-1}}$$

$$E_i = (1 - K_i)^2 \cdot E_{i-1}$$

$$a_i = K_i$$

We could perform this recursion as many times as we needed to compute the desired amount of coefficients. We wrote a MATLAB program to perform the algorithm N times on the windowed signal to generate N coefficients. We used these coefficients in the feedback branches of the filter. We found that we could accurately model the system using a linear prediction filter with 50 coefficients. The frequency response of this filter has a similar shape to the measured frequency response of the system:

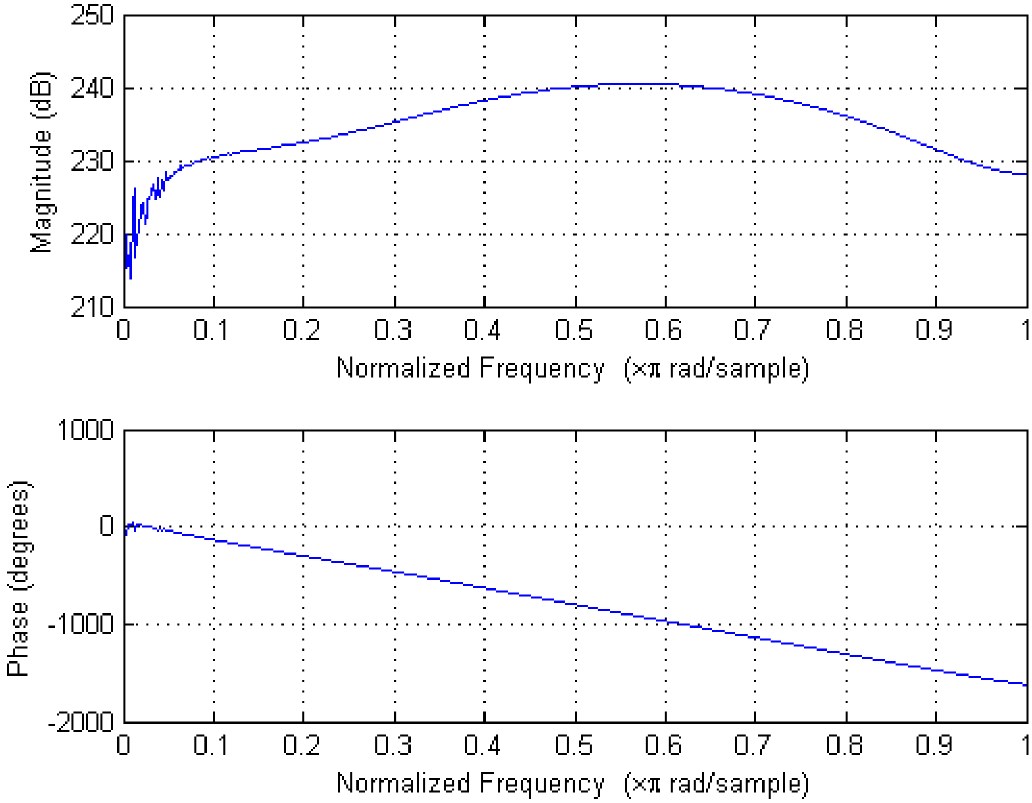


Step 3: Build the Inverse Filter

Step 3: Build the Inverse Filter

The linear prediction filter is simple to invert. Since it uses only the previous outputs to generate the next output, it is an all-pole filter with only feedback branches. To build the inverse filter, we used all the feedback

coefficients that we generated using Burden's Algorithm as the feed-forward coefficients of the inverse filter. The frequency response of the inverse filter looks like:



We observed that the inverse filter accurately inverted the response of the system. It successfully attenuated the low-frequency window vibrations, and it amplified the higher frequencies that the system attenuated.

Vocal Band Pass Filter

After the inverse filter, we decided to isolate the speech signal to remove some of the additive noise. We accomplished this by applying a band pass filter to the recorded signal. When filtering signals, it is very useful to have an understanding of where the important information in the signal lies. With a speech signal there are a few things that we can take advantage of when attempting to filter out noise.

Speech signals generally have a distinctive envelope in the frequency domain (pictured below). After our preliminary filters, we were able to use this envelope to check and see if our output matched.

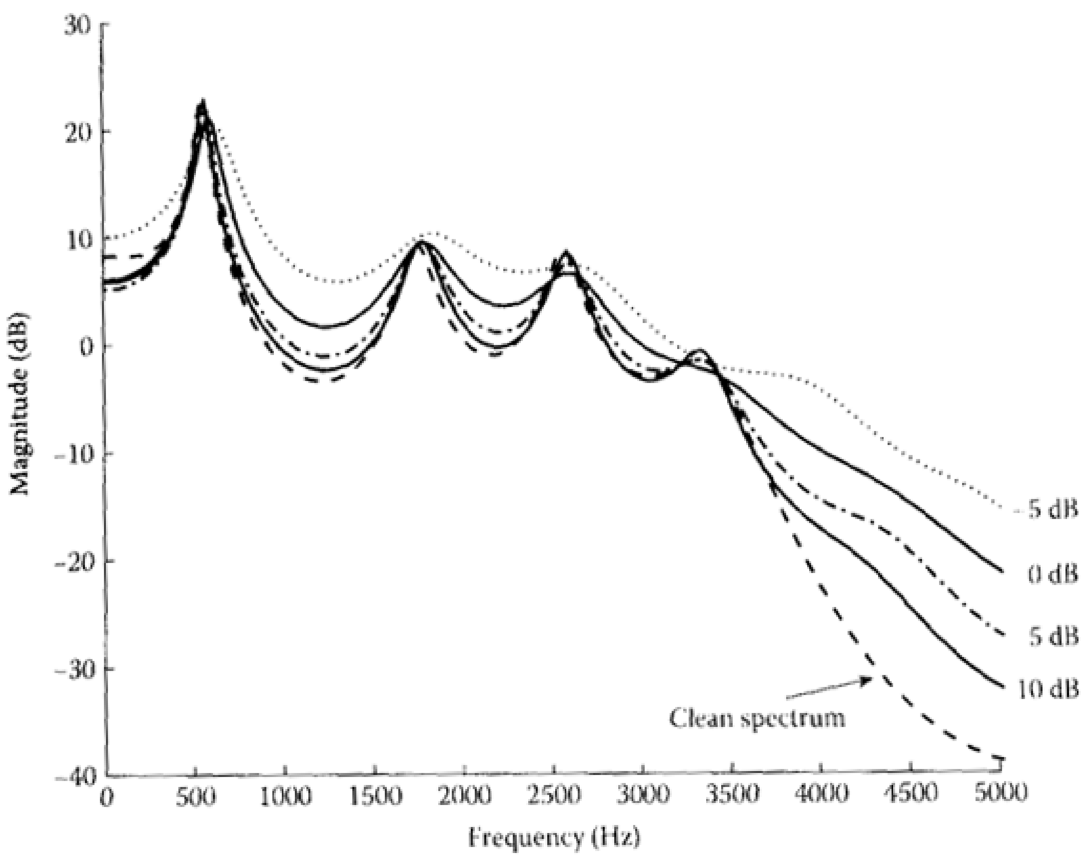
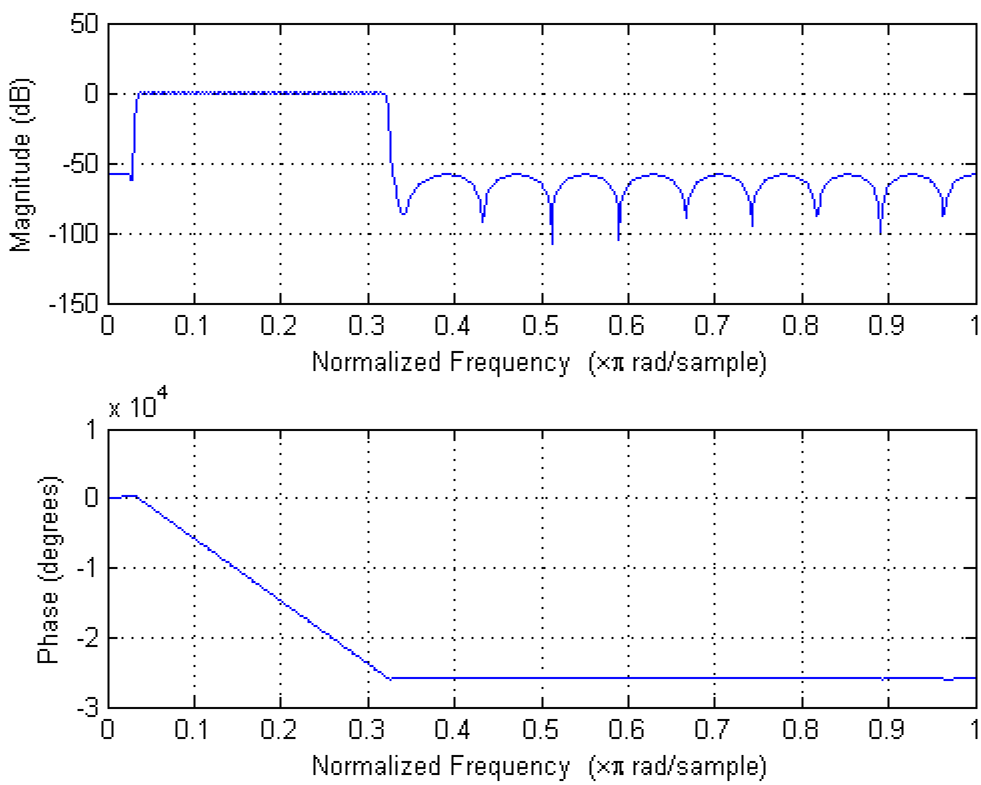


FIGURE 4.7 LPC spectra of the vowel /eh/ (same as in Figure 4.6) corrupted by multi-talker babble at −5 to 10 dB S/N.

Picture from "Speech Enhancement Theory and Practice" Philipos C.

Human speech exists within a finite frequency range. As we are trying to eliminate noise to create a more intelligible speech signal we can get rid of everything outside of this range. To do this we will use a band-pass filter. To get optimum intelligibility telephone companies will generally use a window from 300Hz-3600Hz. The military uses around 400Hz-2800Hz to get rid of more background noise. We used a band-pass filter that went from 400Hz-3600Hz. In order to efficiently design this filter to have linear phase and a finite impulse response, we utilized the Remez Exchange (or Parks McClellan) algorithm. We accomplished this in MATLAB, resulting in the frequency response shown below.

Frequency Response of Bandpass Filter



Results

Both the inverse filter and the vocal band filter performed well at improving the quality of the transmitted signal by compensating for the observed distortion and removing additive noise. The inverse filter successfully boosted the high frequencies that were absorbed by the window. The vocal band filter isolated the speech portion of the signal and successfully removed much of the noise produced by the low-frequency window vibrations. The spectrum of the filtered signal appears similar in shape to the human voice spectrum in the pass band.

Possible Improvements

In order to improve the quality of the recorded signal, we'd like to explore ways of improving the transmission process to get better results. One method that we conceived is to modulate the laser beam at its source with a carrier frequency. We could then demodulate the recorded signal digitally. In theory, this scheme could considerably reduce the amount of additive noise in the transmitted signal by moving the transmitted speech band away from the strong low-frequency noise and into the high-frequency range.

Conclusion

This project was an enjoyable experience for all the members in our group. We got to experiment with a technology that was new to us, and we got to learn a lot about digital speech processing. Overall we are proud of the project.